Задание по программированию: Вторичный индекс в базе данных

Повторить попытку · 0/1 баллов получено
Для успешной сдачи вам необходимо набрать 1/1 баллов.

Срок сдачи Сдайте это задание до May 12, 11:59 PM PDT

Инструкции

Моя работа

Обсуждения

Часто от хранилища данных требуется быстрый поиск по любому полю. Для этого конструируются вспомогательные структуры данных, ускоряющие поиск, они называются вторичными индексами (secondary index). Реализуйте такое хранилище для данных типа Record ниже:

```
struct Record {
  string id;
  string title;
  string user;
  int timestamp;
  int karma;
};

class Database {
public:
  bool Put(const Record& record);
  const Record* GetById(const string& id) const;
  bool Erase(const string& id);

  template <typename Callback>
  void RangeByTimestamp(int low, int high, Callback callback) const;

  template <typename Callback>
  void RangeByKarma(int low, int high, Callback callback) const;

  template <typename Callback>
  void AllByUser(const string& user, Callback callback) const;
};

```

Требования:

- Операция Put должна возвращать true, если вставка удалась, и false, если вставка не удалась, потому что в базе данных уже есть запись с таким же id. В последнем случае состояние базы данных не должно меняться.

- Операция GetById должна возвращать nullptr, если в базе данных нет записи с указанным id.
- Операция Erase должна возвращать true, если удалось удалить элемент с заданным id, и false, если элемент с заданным id не был найден. В последнем случае состояние базы данных не должно меняться.
- Подразумевается, что callback должен возвращать true, если требуется продолжить обход найденных записей, и false в противном случае. Например, это позволит вывести первые 10 записей или найти первую запись, удовлетворяющую дополнительному критерию.
- Время выполнения всех операций должно иметь в среднем сублинейную (например, константную или логарифмическую) зависимость от общего числа записей в базе данных.
- Все границы интервалов - включительные, [low, high] — это наиболее универсальный интервальный запрос, подумайте, как с помощью него выразить условия вида x < high, x > low, x = x_0.
- При решении этой задачи вы можете считать, что bad_alloc кидаться не будет.

Заготовка решения

secondary_index.cpp

Примечание

В этой задаче может оказаться полезен контейнер multimap. Мы не рассматривали его в лекциях, однако вам не составит труда в нём разобраться — он отличается от обычного map только тем, что может хранить повторяющиеся ключи.

Подсказки

Для этой задачи есть набор подсказок, которые должны помочь вам с решением. Если вам не удаётся решить задачу и вы чувствуете, что у вас кончились идеи, вы можете ими воспользоваться. Но сначала обязательно попробуйте решить задачу без подсказок.

secondary_index_hint_1.pdf

secondary_index_hint_2.pdf

secondary_index_hint_3.pdf

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.